AT&T Technical Journal
Vol. 64, No. 7, September 1985
Printed in U.S.A.

A Large-Scale Distribution and Location Model

By J. G. KLINCEWICZ*

(Manuscript received February 25, 1985)

A large-scale, single-period, mathematical programming model of a multicommodity distribution system has been designed and implemented to analyze and to help reconfigure AT&T distribution facilities. Within this model, the number, size, and location of major stocking locations and subsidiary stocking locations are determined on the basis of various incurred costs. These costs include facility setup costs, facility closing costs, and shipping, inventorying, handling, and operating costs. The model incorporates various features that do not appear in standard facility location models, such as nonlinear economies of scale in operating cost, capacity constraints, special products that are handled by only a limited number of facilities, and establishment of subsidiary stocking locations where desirable. In this paper we describe the model, provide a mathematical programming formulation of the problem, and describe the algorithm that was developed to obtain good solutions in an efficient manner. The flexibility of the formulation and the efficiency of the solution technique make this model a unique and useful tool. It can provide insight when used to study an existing or proposed distribution system, and it has already been used in a variety of case studies.

I. INTRODUCTION

Large manufacturing and industrial concerns, such as AT&T, provide for the warehousing and distribution of finished goods. Decisions concerning the number, size, and locations of warehousing and distribution facilities greatly affect the cost of a large material logistics system.

* AT&T Bell Laboratories.

This paper describes a large-scale, single-period, mathematical programming model of a multicommodity distribution system. Various quantitative studies of distribution problems can be performed using this model. For example, an analyst can examine the trade-off between many small distribution facilities and a few large facilities, determine a consolidation strategy in areas of contracting demand, plan an expansion strategy in areas of increasing demand, and so forth.

This model currently is being used to analyze and help reconfigure AT&T distribution facilities in order to meet future material logistics requirements. The flexibility of the model has allowed it to be applied in a variety of in-depth case studies. These studies have included both national and regional studies, studies of different tiers within the distribution system, and studies involving different families of products. To provide this flexibility, components of the model are described in generic terms with a minimum of restrictive assumptions.

Within the model, the number, size, and location of major stocking facilities, called Distribution Centers (DCs), and subsidiary stocking facilities, called Local Distribution Centers (LDCs), are determined on the basis of various incurred costs, including facility setup costs, facility closing costs, and shipping, inventorying, handling, and operating costs. The model is quite complex and combines several features, or combinations of features, that do not appear in standard facility location models. Among these features are nonlinear economies of scale in operating costs, capacity constraints, special products that are handled by only a limited number of facilities, and establishment of subsidiary stocking locations where desirable. These features are discussed in Section II.

The generic distribution system that is considered is illustrated in Fig. 1. We describe this system briefly here; it is described in greater detail in Section II. Within the model, products are assumed to move from various vendors (and repair shops) to the DCs. These DCs, in turn, distribute these products to demand area locations. In certain instances, a group of demand areas can also be served by an LDC. Different products move according to different patterns among vendors, DCs, LDCs and demand areas; these different product "types" are described in particular in Section 2.2. The mathematical programming formulation of this problem is quite large; for example, a problem with 50 possible major distribution center locations, 20 subsidiary stocking locations, 100 demand areas, and 1 special product would involve 13,170 variables and 110,531 constraints. A sample problem of this size was solved in 115.6 CPU seconds on an Amdahl 470/V8 computer.

An extensive literature exists on operations research techniques for discrete facility location problems. A recent survey of facility location

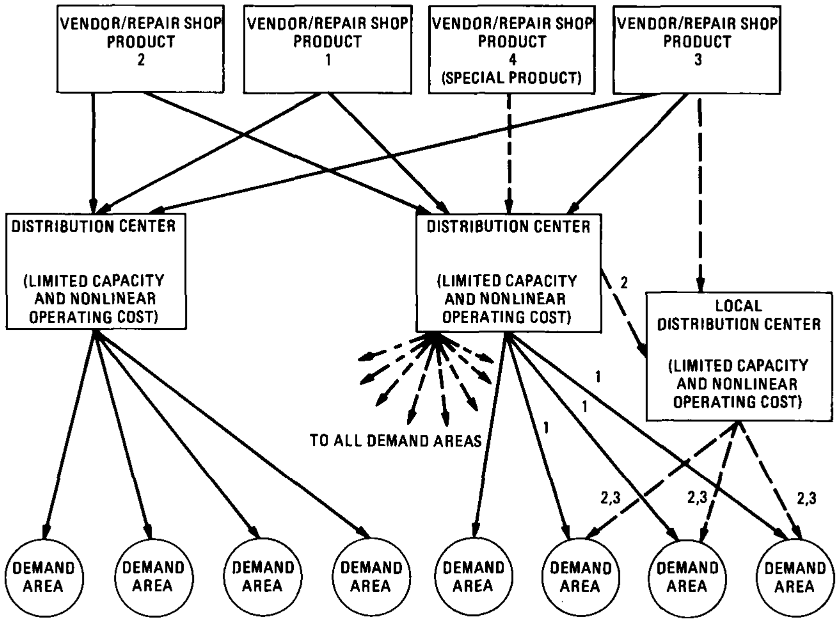

Fig. 1—The advanced distribution and location model.

work is given in Francis, McGinnis, and White.[1] We will here only mention briefly some representative problems and papers.

In the p-median problem, p facilities are chosen so as to minimize the sum of the distances from facilities to customers. A Lagrangian relaxation technique for a capacitated p-median problem is incorporated into the algorithm described in this paper (see Section 3.4). Surveys of work on p-median and related network location subjects can be found in Refs. 2 and 3.

A classic model is the Uncapacitated Facility Location Problem (UFLP), in which there exists a trade-off between setup costs of facilities and the costs of shipping products to demand areas.[4-7] Complete surveys of work on UFLP can be found in Refs. 8 and 9.

Some facility location problems can be modeled as generalized assignment problems (see Ref. 10). Here, customers with demands must be allocated among given facilities with limited capacity. Each customer must be served by only one facility. Lagrangian relaxation techniques for this problem are discussed in Fisher[11] and Ross and Soland.[10] A related procedure is incorporated into the algorithm described here in Section 3.2.

The Capacitated Facility Location Problem (CFLP)—in which a customer's demand can be split among several facilities—is considered, for example, in Refs. 12, 13, and 14. Some authors have considered the case in which there are concave operating costs associated with each facility. See, for example, Refs. 15, 16, and 17. In particular, the algorithm of Kelly and Khumawala,[15] which involves solving a sequence of linear transportation problems, is adapted here for use in the major optimization routine (see Section 3.3).

A Benders decomposition approach for a multicommodity problem was developed by Geoffrion and Graves.[18] Discussion of dynamic facility location can be found in Erlenkotter[19] and in a section of the capacity expansion survey by Luss.[20]

In Section II, we provide a complete description of the material logistics system being modeled and the mathematical programming formulation of the problem. Section III describes the solution technique that was implemented. The first subsection provides a general overview of the various stages of the algorithm. The subsequent subsections describe each stage in turn. Some brief discussion of implementation details and some concluding remarks are then given in Section IV.

II. DESCRIPTION OF THE PROBLEM

2.1 Types of locations and facilities

Five major types of facilities or physical locations can be identified in the material logistics system. Several products $k = 1, \cdots, K$ move among these facilities and locations.

Vendors are those locations, such as factories and manufacturing locations, from which products first enter the material logistics system. Each product k has its own set of fixed and known vendor locations $n = 1, \cdots, N_k$.

Repair shops are similar to vendors. Each product has its own set of fixed and known repair shop locations $m = 1, \cdots, M_k$. For each product, some fraction of demand ρ_k (possibly zero) is to be satisfied by repaired items.

Demand areas are geographical areas to which products are ultimately destined. These demand areas $j = 1, \cdots, J$, and the amount of demand D_{jk} for each product k at each demand area j, are assumed to be fixed and known.

Distribution centers are major intermediate stocking locations. The DCs must be chosen from among a set of locations $i = 1, \cdots, I$, which can be either "potential" or "existing." For potential locations, the model specifies a minimum capacity size B_i^1, a feasible capacity increment b_i^1, and a maximum capacity size $\bar{B}_i^1$. For existing facilities with fixed capacity, we assume $B_i^1 = \bar{B}_i^1$ and $b_i^1 = 0$. DC i can serve demand areas within a radius of δ_i^1 miles.

Local distribution centers are subsidiary stocking locations that handle certain types of products for several nearby demand areas. Depending upon the product type, LDCs receive products from either a DC or directly from vendors and repair shops; they then ship the products to demand areas. LDC locations must be chosen from among a set of existing and potential locations $\ell = 1, \cdots, L$. Potential facilities have minimum capacity B_ℓ^2, maximum capacity $\bar{B}_\ell^2$, and capacity increment b_ℓ^2. Existing facilities have fixed capacity $B_\ell^2 = \bar{B}_\ell^2$ and $b_\ell^2 = 0$. An LDC ℓ can serve demand areas within a radius of δ_ℓ^2 miles.

Ordinarily, each DC deals only with certain demand areas that are "assigned" to it. The DC is (perhaps) associated with some LDCs. Each demand area that is assigned to a given DC obtains products only from that DC, and from at most one of the LDCs associated with it. Some products are exceptions to this rule; they are discussed in Section 2.2.

2.2 Types of products

Many different products move through this material logistics system. For modeling purposes, each "product" may represent an aggregation of several products. We distinguish four categories or types of products, according to how they are handled within the system. These types are described below and are pictured in Fig. 1. In Fig. 1, the three demand areas in the lower right corner are assigned to an LDC as well as a DC, whereas the other demand areas are assigned only to a DC.

Type 1 products can be handled only by a DC. These products are shipped to the DC from the vendor and repair shop. (We assume that the particular vendor and repair shop that supply a given DC are chosen so as to minimize shipping cost.) From the DC, they are shipped to all demand areas that are assigned to that DC.

Type 2 products can be delivered to demand areas either from a DC or else from an LDC. If a demand area is assigned to an LDC, the DC ships Type 2 products to the LDC for delivery to the demand area. This is indicated by the dashed lines in Fig. 1 that are labeled with the numeral 2.

Type 3 products can also be delivered to demand areas either from a DC or else from an LDC. The LDC receives shipment of Type 3 products directly from the vendor and repair shop. We assume that the LDC receives shipments from the same repair shops and vendors that serve the DC with which the LDC is associated. This is indicated by the dashed lines in Fig. 1 that are labeled with the numeral 3.

All Type 1, Type 2 and Type 3 products are handled with the arrangement whereby each demand area deals with only one DC and at most one LDC. (If a demand area is assigned to an LDC, it obtains all Type 2 and Type 3 products via that LDC.) However, Type 4 products ("special" products) are an exception to the rule. Only a small number p_k of DCs are chosen to handle a given Type 4 product k. These DCs then serve all demand areas. This is indicated by the alternate short-and-long dashed lines in Fig. 1.

Each product's type is given as input by the user. We let T_1, T_2, T_3, and T_4 denote the sets of indices of Type 1, Type 2, Type 3, and Type 4 products, respectively.

Each unit of demand that is handled by a DC or an LDC occupies some average amount of warehouse space (measured in units of capacity). The amount of space occupied depends upon, among other things, the size of the product, turnover time of inventory, and the type of facility (DC or LDC). We define the following parameters:

s_k^1 = warehouse space required per unit demand at a DC if a DC ships product k to a demand area,
s_k^2 = warehouse space required per unit demand at an LDC,
s_k^3 = warehouse space required per unit demand at a DC if a DC ships product k to an LDC (relevant for Type 2 products).

2.3 Costs

2.3.1 Facility setup costs

Facility setup costs are costs incurred when a facility (DC or LDC) is chosen to be open. The setup cost depends upon the size of the facility. We assume that, for DC i, opened with a capacity of x units, the setup cost is of the form $\alpha_i^1 + \beta_i^1 x$, where α_i^1 and β_i^1 are given constants that depend upon the particular facility. For LDC ℓ, the setup cost is of the form $\alpha_\ell^2 + \beta_\ell^2 x$, where α_ℓ^2 and β_ℓ^2 are given constants. Actual total setup costs realistically might be assumed to be amortized over several time periods. Since ours is a static, single-period model, the setup cost used in the model could be set equal to an amortized share of the total setup cost.

2.3.2 Facility closing costs

In the model, if a facility is not chosen to be open, a closing cost c_i^1 (for DC i) or c_ℓ^2 (for LDC ℓ) is incurred.

2.3.3 Shipping costs

Shipping cost parameters are given in cost per mile per unit demand. These cost parameters for product k follow:

t_k^1 = cost of shipping from vendor/repair shop to DC/LDC,
t_k^2 = cost of shipping from DC to demand area,
t_k^3 = cost of shipping from DC to LDC,
t_k^4 = the cost of shipping from LDC to demand area.

Obviously, for each type of product only certain of these parameters are applicable. (The model can also be easily modified so that shipping costs are measured per unit demand, independent of distance.)

2.3.4 Inventorying or storage cost

An inventorying or storage cost per unit demand is incurred at each DC and LDC. We define the following parameters:

η^1_{ki} = inventorying cost per unit demand at a DC i that ships product k to a demand area,
$\eta^2_{k\ell}$ = inventorying cost per unit demand at an LDC ℓ that ships product k to a demand area,
η^3_{ki} = inventorying cost per unit demand at a DC i that ships product k to an LDC (relevant for Type 2 products).

The value of η^3_{ki} can differ from η^1_{ki} because of differences in turnover time for inventory bound for an LDC and inventory bound directly for demand areas.

2.3.5 Handling cost

When a product is processed at a warehousing facility (either a DC or an LDC), there are some labor costs incurred. We define the following parameters:

h^1_{ki} = handling cost per unit demand at a DC i that ships product k to a demand area,
$h^2_{k\ell}$ = handling cost per unit demand at an LDC ℓ that ships product k to a demand area,
h^3_{ki} = handling cost per unit demand at a DC i that ships product k to an LDC ℓ (relevant for Type 2 products).

2.3.6 Operating cost

Other operating costs at each facility are represented as a continuous, nondecreasing, concave function of the space occupied in order to account for possible savings due to economies of scale. Let

σ^1_i = volume of space occupied at DC i, $i = 1, \cdots, I$, and
σ^2_ℓ = volume of space occupied at LDC ℓ, $\ell = 1, \cdots, L$.

Then, we define the cost functions:

$f^1_i(\sigma^1_i)$ = operating cost at DC i, $i = 1, \cdots, I$, and
$f^2_\ell(\sigma^2_\ell)$ = operating cost at LDC ℓ, $\ell = 1, \cdots, L$.

Within the software implementation, we assume these cost functions to be piecewise linear with a nonnegative intercept at the origin. In typical examples, each function used between three and five piecewise linear segments.

2.4 The mathematical programming model

2.4.1 Decision variables

To formulate the mathematical programming model, we first specify the necessary decision variables. All variables take on integer values. They are as follows:

$$y_i = \begin{cases} 1 \text{ if DC } i \text{ is open,} \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, \cdots, I,$$

$$z_{i\ell} = \begin{cases} 1 \text{ if LDC } \ell \text{ is open and served by DC } i, \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } \ell = 1, \cdots, L \quad \text{and} \quad i = 1, \cdots, I,$$

$$x_{ij} = \begin{cases} 1 \text{ if demand area } j \text{ is assigned to DC } i, \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, \cdots, I \quad \text{and} \quad j = 1, \cdots, J,$$

$$w_{\ell j} = \begin{cases} 1 \text{ if demand area } j \text{ is assigned to LDC } \ell, \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } \ell = 1, \cdots, L \quad \text{and} \quad j = 1, \cdots, J,$$

$$v_{ik} = \begin{cases} 1 \text{ if DC } i \text{ is used to serve special product } k, \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, \cdots, I \quad \text{and} \quad k \in T_4,$$

$$u_{ijk} = \begin{cases} 1 \text{ if demand area } j \text{ receives special product } k \text{ from DC } i, \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } j = 1, \cdots, J \quad \text{and} \quad i = 1, \cdots, I \quad \text{and} \quad k \in T_4,$$

$$q_i^1 = \text{number of size increments above the minimum opened at DC } i, \text{ for } i = 1, \cdots, I,$$

$$q_\ell^2 = \text{number of size increments above the minimum opened at LDC } \ell, \text{ for } \ell = 1, \cdots, L.$$

2.4.2 Distance and other parameters

Shipping cost calculations require the distances between pairs of locations. For notational simplicity, we refer to all such pairwise distances by the notation d with two subscripts. The subscript i refers to a DC i, subscript ℓ to an LDC ℓ, and subscript j to a demand area j. Further, the subscript $n(i)$ refers to the vendor closest to DC i, and subscript $m(i)$ refers to the repair shop closest to DC i. Thus, $d_{n(i),i}$ is the distance to DC i from its nearest vendor, $d_{\ell j}$ is the distance from LDC ℓ to demand area j, and so forth.

A particular latitude/longitude point is associated with each facility location and each demand area and used to estimate road distances on an as-needed basis. The software allows users to provide distance data that would override the calculated distance.

To simplify the formulation, let the space (units of capacity) required for a DC to serve demand area j (excluding Type 4 products and provided it is not served by an LDC) be

$$S_j^1 = \sum_{k \in T_1 \cup T_2 \cup T_3} s_k^1 D_{jk}. \tag{1}$$

If demand area j is served by an LDC as well as a DC, then the amount of space required at the LDC is

$$S_j^2 = \sum_{k \in T_2 \cup T_3} s_k^2 D_{jk}, \tag{2}$$

and the amount of space required at the DC is

$$S_j^3 = \sum_{k \in T_1} s_k^1 D_{jk} + \sum_{k \in T_2} s_k^3 D_{jk}. \tag{3}$$

The space required at a DC to serve Type 4 product $k \in T_4$ for demand area j is

$$S_{jk}^4 = s_k^1 D_{jk}. \tag{4}$$

Likewise, it is convenient to aggregate the shipping costs, inventorying costs, and handling costs associated with assigning demand area j to DC i in an "assignment cost" A_{ij}^1, as follows:

$$A_{ij}^1 = \sum_{k \in T_1 \cup T_2 \cup T_3} (t_k^1 d_{n(i)i}(1 - \rho_k) + t_k^1 d_{m(i)i} \rho_k + t_k^2 d_{ij} + h_{ki}^1 + \eta_{ki}^1) D_{jk}. \tag{5}$$

If an LDC ℓ is involved, the assignment cost $A_{ij\ell}^2$ is expressed as

$$\begin{aligned} A_{ij\ell}^2 = & \sum_{k \in T_1} (t_k^1 d_{n(i)i}(1 - \rho_k) + t_k^1 d_{m(i)i} \rho_k + t_k^2 d_{ij} + h_{ki}^1 + \eta_{ki}^1) D_{jk} \\ & + \sum_{k \in T_2} (t_k^1 d_{n(i)i}(1 - \rho_k) + t_k^1 d_{m(i)i} \rho_k + t_k^3 d_{i\ell} + t_k^4 d_{\ell j} \\ & + h_{ki}^3 + h_{k\ell}^2 + \eta_{ki}^3 + \eta_{k\ell}^2) D_{jk} \\ & + \sum_{k \in T_3} (t_k^1 d_{n(i)\ell}(1 - \rho_k) + t_k^1 d_{m(i)\ell} \rho_k \\ & + t_k^4 d_{\ell j} + h_{k\ell}^2 + \eta_{k\ell}^2) D_{jk}. \end{aligned} \tag{6}$$

Finally, the cost A_{ijk}^4 of assigning Type 4 product k at demand area j to DC i is as follows:

$$A_{ijk}^4 = (t_k^1 d_{n(i)i}(1 - \rho_k) + t_k^1 d_{m(i)i} \rho_k + t_k^2 d_{ij} + h_{ki}^1 + \eta_{ki}^1) D_{jk}. \tag{7}$$

In the event that demand area j cannot be assigned to a DC i or an LDC ℓ because j lies outside the operating radius of the facility (i.e., $d_{ij} > \delta_i$ or $d_{\ell j} > \delta_\ell$), then the corresponding assignment costs can be set to an arbitrarily large number.

2.4.3 Formulating the model

Below, we provide the mathematical formulation for the problem. Then, the objective function and each of the constraints is explained in turn. The problem is formulated as follows:

$$\begin{aligned}
\min \sum_{i=1}^{I} & (\alpha_i^1 + \beta_i^1(B_i^1 + b_i^1 q_i^1)) y_i \\
& + \sum_{\ell=1}^{L} (\alpha_\ell^2 + \beta_\ell^2(B_\ell^2 + b_\ell^2 q_\ell^2)) \sum_{i=1}^{I} z_{i\ell} \\
& + \sum_{i=1}^{I} c_i^1(1 - y_i) + \sum_{\ell=1}^{L} c_\ell^2 \left(1 - \sum_{i=1}^{I} z_{i\ell}\right) \\
& + \sum_{i=1}^{I} \sum_{j=1}^{J} A_{ij}^1 x_{ij} \left(1 - \sum_{\ell=1}^{L} w_{\ell j}\right) \\
& + \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{\ell=1}^{L} A_{ij\ell}^2 x_{ij} w_{\ell j} \\
& + \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k \in T_4} A_{ijk}^4 u_{ijk} \\
& + \sum_{i=1}^{I} f_i^1(\sigma_i^1) y_i \\
& + \sum_{\ell=1}^{L} f_\ell^2(\sigma_\ell^2) \sum_{i=1}^{I} z_{i\ell} \qquad (8a)
\end{aligned}$$

subject to

$$\sum_{i=1}^{I} x_{ij} = 1 \quad \text{for} \quad j = 1, \cdots, J, \qquad (8b)$$

$$x_{ij} \leq y_i \quad \text{for} \quad i = 1, \cdots, I \quad \text{and} \quad j = 1, \cdots, J, \qquad (8c)$$

$$\sum_{\ell=1}^{L} w_{\ell j} \leq 1 \quad \text{for} \quad j = 1, \cdots, J, \qquad (8d)$$

$$w_{\ell j} \leq x_{ij} z_{i\ell} \quad \text{for} \quad i = 1, \cdots, I \quad \text{and} \quad \ell = 1, \cdots, L \quad \text{and} \quad j = 1, \cdots, J, \qquad (8e)$$

$$\sum_{i=1}^{I} z_{i\ell} \leq 1 \quad \text{for} \quad \ell = 1, \cdots, L, \qquad (8f)$$

$$\sigma_i^1 = \sum_{j=1}^{J} S_j^1 x_{ij} \left(1 - \sum_{\ell=1}^{L} w_{\ell j}\right) + \sum_{\ell=1}^{L} \sum_{j=1}^{J} S_j^3 x_{ij} w_{\ell j} + \sum_{k \in T_4} \sum_{j=1}^{J} S_{jk}^4 u_{ijk} \quad \text{for} \quad i = 1, \cdots, I, \tag{8g}$$

$$\sigma_i^1 \leq B_i^1 + b_i^1 q_i^1 \quad \text{for} \quad i = 1, \cdots, I, \tag{8h}$$

$$\sigma_\ell^2 = \sum_{j=1}^{J} S_j^2 w_{\ell j} \quad \text{for} \quad \ell = 1, \cdots, L, \tag{8i}$$

$$\sigma_\ell^2 \leq B_\ell^2 + b_\ell^2 q_\ell^2 \quad \text{for} \quad \ell = 1, \cdots, L, \tag{8j}$$

$$B_i^1 + b_i^1 q_i^1 \leq \bar{B}_i^1 \quad \text{for} \quad i = 1, \cdots, I, \tag{8k}$$

$$B_\ell^2 + b_\ell^2 q_\ell^2 \leq \bar{B}_\ell^2 \quad \text{for} \quad \ell = 1, \cdots, L, \tag{8l}$$

$$\sum_i v_{ik} \leq \rho_k \quad \text{for} \quad k \in T_4, \tag{8m}$$

$$u_{ijk} \leq v_{ik} \quad \text{for} \quad j = 1, \cdots, J, \text{ and } i = 1, \cdots, I \text{ and } k \in T_4, \tag{8n}$$

$$\sum_i u_{ijk} = 1 \quad \text{for} \quad j = 1, \cdots, J \text{ and } k \in T_4, \tag{8o}$$

$$y_i, z_{i\ell}, x_{ij}, w_{\ell j}, v_{ik}, u_{ijk} \in \{0, 1\} \text{ for } i = 1, \cdots, I, \; j = 1, \cdots, J, \; \ell = 1, \cdots, L, \text{ and } k \in T_4, \tag{8p}$$

$$q_i^1, q_\ell^2 \in \{0, 1, \cdots, \infty\} \quad \text{for} \quad i = 1, \cdots, I \text{ and } \ell = 1, \cdots, L. \tag{8q}$$

The first two summation terms in the objective function (8a) represent the setup cost incurred for open DCs and LDCs. The next two terms represent the closing costs that are incurred if a DC or an LDC is not open. In the fifth term, we include the assignment costs that are incurred if demand area j is assigned to DC i with no LDC involved. (Only in that case would x_{ij} $(1 - \sum_{\ell=1}^{L} w_{\ell j}) = 1$.) The sixth term, on the other hand, gives the assignment costs that are incurred if demand area j is assigned to DC i and LDC ℓ, and the seventh term considers the assignment costs for Type 4 products. The last two terms of the objective function represent the operating cost that is incurred at each open facility.

Constraints (8b) ensure that each demand area j is assigned to exactly one DC, and constraints (8c) ensure that demand areas are assigned only to DCs that are open. Constraints (8d) permit each demand area j to be assigned to at most one LDC. The condition (8e) guarantees that such an LDC assignment is made only if the LDC is open and is served by the same DC that serves the demand area j. Constraints (8f) ensure that each LDC is served by at most one DC. Constraints (8g) define σ_i^1, the space actually utilized at each DC i. This is obtained by adding the space required to serve demand areas that are assigned only to DC i, plus space required for demand areas assigned to both DC i and an LDC, plus space required to serve any Type 4 products that are assigned to DC i. In (8h), this space utilized is constrained to not exceed the capacity installed. Similarly, (8i) defines σ_ℓ^2, the space utilized at LDC ℓ, and (8j) constrains σ_ℓ^2 to not exceed the capacity installed at that LDC. Constraints (8k) and (8l) ensure that the capacity installed does not exceed the maximum permitted capacity for DCs and LDCs, respectively. In (8m), the number of DCs that serve each Type 4 product k does not exceed the permitted number p_k. Constraints (8n) guarantee that a demand area receives each Type 4 product k from a DC that handles that product. Constraints (8o) require that each demand area j be assigned to only one DC for a given Type 4 product. [In the event that no Type 4 products occur in the problem, constraints (8m) through (8o) do not appear]. Finally, conditions (8p) and (8q) enforce integer constraints on the variables.

The integer program (8) is quite large. For example, for 10 possible DC locations, 10 possible LDC locations, 50 demand areas, and 1 Type 4 product, there are 1640 variables and 6221 constraints. For a larger problem of 50 DCs, 20 LDCs, 100 demand areas, and 1 Type 4 product, there are 13,170 variables and 110,531 constraints. Nonlinearities appear in the objective function (8a) and in constraints (8e) and (8g), thus making it even more difficult to solve the program directly.

III. SOLUTION APPROACH

3.1 Overview

Because of the difficulty of the integer programming problem, the proposed algorithm contains some heuristic elements. In particular, we propose to first treat a simpler version of the problem and then adjust this solution with a series of heuristics to obtain a solution to the overall problem.

The elements of the problem that are judged to be most important are considered in the initial optimization. Other elements that are considered, by comparison, less important or elements that are expected to appear less often in actual case studies are treated by the secondary optimization. Specifically, the issues of Type 4 product distribution, LDC locations, and discrete facility sizing are set aside in the initial optimization. The initial optimization problem thereby becomes a type of capacitated facility location problem with concave costs. After obtaining a solution to this problem, the solution is modified in a step-by-step fashion to incorporate, in turn, Type 4 products, LDC locations and facility sizing. In the subsections that follow, we describe the various portions of the algorithm.

Within our software implementation, there is the option to specify that certain variables be fixed in advance, for example, that certain DCs or LDCs be fixed open or closed, that certain demand area assignments be forced or forbidden, or that certain DCs be prevented from handling Type 4 products. The necessary modifications to the algorithm are generally straightforward, and therefore not discussed explicitly here.

3.2 The preprocessor

In Section 2.4.2 we described various DC and LDC space requirement parameters and assignment cost parameters. These parameters are computed by the algorithm in a preprocessor routine. The importance of the space requirement parameters lies in the fact that, when not considering Type 4 products, the original *multicommodity* problem becomes a *single-commodity* problem. The "commodity" in this case is warehouse space; the DCs have supplies of space and the demand areas require space. Further, the assignment costs (5) and (7) provide a convenient aggregation of shipping, inventorying, and handling costs. (Because of storage requirements, however, the assignment cost for assignments that use LDCs [eq. (6)] is calculated as needed.)

3.3 The primary optimization

In the primary optimization routine, we determine a set of DCs to be opened. Initially, we assume that all demand is served by DCs alone and that the maximum possible capacity $\bar{B}_i$ is available at all open facilities. We associate a cost with each possible pairing of a DC i and a demand area j of the form

$$C_{ij} = A^1_{ij} + \beta^1_i S^1_j. \tag{9}$$

This represents the assignment cost for demand area j (as discussed in Section 2.4.2) plus a cost corresponding to the variable setup cost for the warehouse space required to serve j. We also compute a "net fixed cost"

$$F_i = \alpha^1_i - c^1_i, \tag{10}$$

which is equal to the difference between the fixed setup cost and the closing cost.

At first, we also set aside the requirement that each demand area be served by only one DC. After first obtaining a solution without this restriction, we then will adjust the solution to enforce the restriction.

The initial problem is then the following capacitated facility location problem with concave operating cost:

$$\min \sum_{i=1}^{I} F_i y_i + \sum_{i=1}^{I} f_i^1(\sigma_i^1) + \sum_{i=1}^{I} \sum_{j=1}^{J} C_{ij} x_{ij} \tag{11a}$$

subject to

$$\sum_{j=1}^{J} S_j^1 x_{ij} \leq \bar{B}_i^1 y_i \quad \text{for} \quad i = 1, \cdots, I, \tag{11b}$$

$$\sum_{i=1}^{I} x_{ij} = 1 \quad \text{for} \quad j = 1, \cdots, J, \tag{11c}$$

$$\sigma_i^1 = \sum_{j=1}^{J} S_j^1 x_{ij} \quad \text{for} \quad i = 1, \cdots, I, \tag{11d}$$

$$0 \leq x_{ij} \leq 1 \quad \text{for} \quad i = 1, \cdots, I \quad \text{and} \quad j = 1, \cdots, J, \tag{11e}$$

$$y_i \in \{0, 1\} \quad \text{for} \quad i = 1, \cdots, I. \tag{11f}$$

Several algorithms for problems of this form have been proposed.[15-17] We have implemented the iterative algorithm due to Kelly and Khumawala[15] that defines and solves a sequence of standard linear transportation problems. In these problems, the DCs are "sources" and demand areas are "sinks." Linear costs on arcs from sources to sinks are based on the values of C_{ij} and on the values of $f_i^1(\sigma_i^1)$ that are implied by the solution at the previous iteration. Key to the algorithm is the provision for a "dummy sink." Incoming arcs to this dummy sink have negative costs based on the setup costs F_i and the operating cost values $f_i^1(\sigma_i^1)$ at the previous iteration. Intuitively, this dummy sink offers to "buy back" the capacity at a DC. Costs on arcs into this sink are chosen from iteration to iteration in such a way as to discourage opening very underutilized DCs and to encourage taking advantage of the economies-of-scale in the concave operating cost. Complete details are given in Ref. 15. Within our implementation, the linear transportation problems are solved using primal network flow-convex,[21] which is a state-of-the-art simplex network flow code.

Upon solving problem (11), we obtain a set G of open DCs, each of which serves some group of demand areas. Some set U of demand areas will be served by more than one DC. Typically, in our computational experience the numbers of such demand areas that have their demand "split" in this way among DCs is small. We next seek to resolve these splits and obtain a solution in which each demand area is served by only one DC.

Within this phase, we momentarily leave fixed those demand areas that are assigned to only one DC in the solution obtained to (11). The demand areas $j \in U$ whose assignments have been split are considered to be unassigned. The fixed assignments, on the other hand, take up warehouse space at the open DCs. The remaining available warehouse space at each DC i is denoted $\tilde{B}_i$.

To take the unassigned demand areas and assign them among the open DCs, we must, if possible, (approximately) solve this problem:

$$\min \sum_{i \in G} \sum_{j \in U} C_{ij} x_{ij} \tag{12a}$$

$$\text{subject to } \sum_{i \in G} x_{ij} = 1 \quad \text{for} \quad j \in U, \tag{12b}$$

$$\sum_{j \in U} S_j^1 x_{ij} \leq \tilde{B}_i \quad \text{for} \quad i \in G, \tag{12c}$$

$$x_{ij} \in \{0, 1\} \quad \text{for} \quad j \in U \quad \text{and} \quad i \in G. \tag{12d}$$

This is of the form of a generalized assignment problem.[10,11] As described below, we first attempt to solve this using Lagrangian relaxation (see Ref. 11) and branch-and-bound techniques. Later, we add additional DCs to the set G in the event that (12) is infeasible.

If we associate nonnegative Lagrange multipliers λ_i, $i \in G$ with constraints (12c) and incorporate these constraints into the objective (12a), we obtain the relaxed problem:

$$\min \sum_{i \in G} \sum_{j \in U} C_{ij} x_{ij} + \sum_{i \in G} \lambda_i \left(\sum_{j \in U} S_j^1 x_{ij} - \tilde{B}_i \right) \tag{13a}$$

subject to

$$\sum_{i \in G} x_{ij} = 1 \quad \text{for} \quad j \in U, \tag{13b}$$

$$x_{ij} \in \{0, 1\} \quad \text{for} \quad i \in G \quad \text{and} \quad j \in U. \tag{13c}$$

The relaxed problem can be solved by inspection. For each j, let index e be chosen so that

$$C_{ej} + \lambda_e S_j^1 = \min_{i \in G} \{C_{ij} + \lambda_i S_j^1\}. \tag{14}$$

(Ties can be broken arbitrarily.) Then, for each demand area j,

$$x_{ij} = \begin{cases} 1 \text{ if } i = e \\ 0 \text{ otherwise} \end{cases} \quad \text{for } i \in G. \tag{15}$$

This is incorporated into an iterative scheme where the λ_i are increased from iteration to iteration if the solution obtained violates constraints (12c). This update is of the form

$$\lambda_i \leftarrow \lambda_i + \tau \left(\sum_{j \in U} S_j^1 x_{ij} - \tilde{B}_i \right), \tag{16}$$

where "←" indicates "is replaced by" and τ is a positive scalar. A formula for τ is

$$\tau = \frac{\gamma Z}{\left\{ \sum_{i=1}^{I} \left\{ \sum_{j=U} S_j^1 x_{ij} - \tilde{B}_i \right\}^2 \right\}^{1/2}}, \qquad (17)$$

where γ is a positive scalar (in our implementation, $\gamma = 0.25$ has been used successfully) and Z is an estimate of the difference between the optimal objective function value (12a) in the original problem and the current value (13a) in the relaxed problem. Within our implementation, we obtain an estimate of the magnitude of (12a) based on arbitrarily assigning each $j \in U$ to one of the DCs to which it has a split assignment. The value of Z is then chosen to be 10 percent of this estimate of (12a).

Similar updating rules appear in Held, Wolfe, and Crowder.[22] Similar relaxation techniques for generalized assignment problems are discussed in Fisher.[11]

If the Lagrangian relaxation problem (13) does not yield a feasible solution to (12) within a reasonable number of iterations (40 iterations have been used successfully in our implementation), then a branch-and-bound technique for (12) is initiated. A description of branch-and-bound algorithms for integer programming problems can be found, for example, in Ref. 23. We outline below the major features of the algorithm that we have implemented.

At each node of the branch-and-bound tree, assignments for some of the demand areas $j \in U$ are assumed fixed. At each node, a Lagrangian relaxation problem, similar to (13), is solved for the unfixed demand areas. A lower bound on the optimal objective function value (12a), given the fixed assignments at the node, is obtained by evaluating the relaxed objective function (13a) at solution (15). Likewise, another lower bound at the node can also be calculated by assuming that each unfixed demand area is assigned to the least costly DC. The branch-and-bound routine uses the maximum of the two lower bounds. If a feasible solution is found by Lagrangian relaxation, it is an upper bound. A very simple branching rule is used in which variables are chosen for branching in order of increasing j. Nodes are chosen for branching by the least lower bound value among most recently created unfathomed nodes. If the lower bound at a given node is within, say, 10 percent of a current upper bound, then no further branching is done from that node.

If problem (12) is not feasible, an additional facility is chosen to be in the open set and then problem (13) is resolved with the DCs in the larger set G fixed open and all other DCs fixed closed. This additional new facility is chosen according to a rule that estimates the change in assignment cost that would result from its opening. That is, for each DC $i \notin G$, compute

$$\Omega_i = \sum_{j \notin U} \min_{k \in G}\{\max(C_{kj} - C_{ij}, O)\}. \tag{18}$$

Each term in this summation gives, for a demand area j, the minimum savings in assignment cost that would result if DC i were available. The DC with maximum Ω_i value is chosen to be open and is thus added to G. (This rule is similar to the "largest omega" rule proposed by Khumawala[6] as a branching rule in a branch-and-bound algorithm for plant location problems.) The Kelly and Khumawala algorithm for problem (11) is then repeated with facilities $i \in G$ open and all others forced to be closed.

In Fig. 2, we provide a flowchart of the primary optimization routine. This figure summarizes the procedures described in this subsection.

3.4 Assigning Type 4 products

As explained in Section 2.2, for each Type 4 product k, only a limited number p_k of DC facilities are chosen to handle it. These facilities can be different for each k. We assume that Type 4 products occupy, at most, only a small percentage (perhaps 10 percent or less) of the warehouse space required.

Given the demand area assignments determined by the primary optimization algorithm, each open DC $i \in G$ only has some amount $\tilde{B}_i$ of warehouse space still available. If demand area j is assigned to DC i for Type 4 product k, we associate cost

$$C^4_{ijk} = A^4_{ijk} + \beta^1_i S^4_{jk}, \tag{19}$$

which represents the assignment cost plus a share of the variable setup cost.

Given the set G, the problem of choosing p_k locations to serve product k can be formulated as the following capacitated p-median problem:

$$\min \sum_{i \in G} \sum_{j=1}^{J} C^4_{ijk} u_{ijk} \tag{20a}$$

subject to

$$\sum_{i \in G} v_{ik} \leq p_k, \tag{20b}$$

$$\sum_{i \in G} u_{ijk} = 1 \quad \text{for} \quad j = 1, \cdots, J, \tag{20c}$$

$$u_{ijk} \leq v_{ik} \quad \text{for} \quad j = 1, \cdots, J \quad \text{and} \quad i \in G, \tag{20d}$$

$$\sum_{j=1}^{J} S^4_{jk} u_{ijk} \leq \tilde{B}_i \quad \text{for} \quad i \in G, \tag{20e}$$

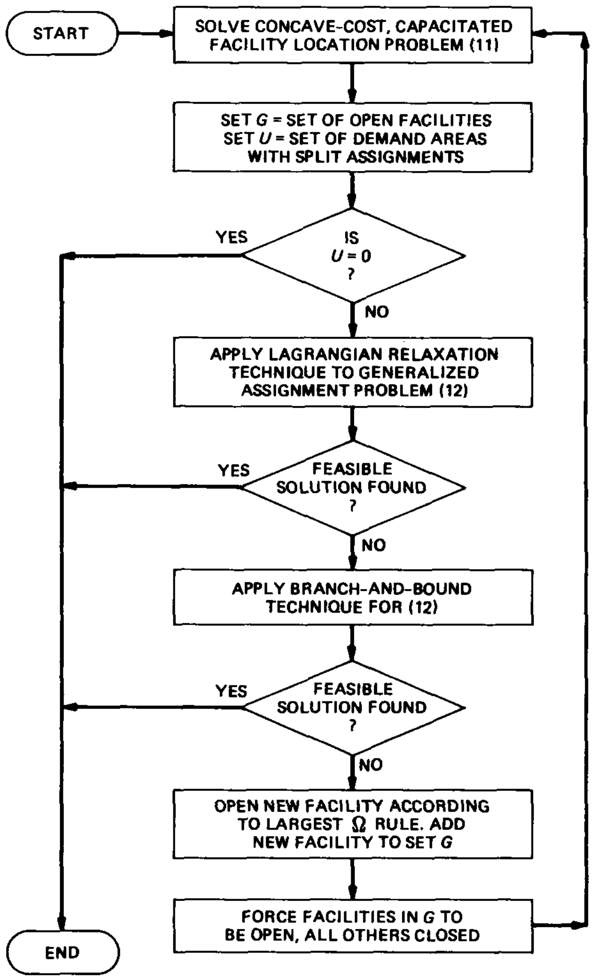

Fig. 2—The primary optimization routine.

$$u_{ijk},\ v_{ik} \in \{0, 1\} \quad \text{for} \quad j = 1, \cdots, J \quad \text{and} \quad i \in G. \tag{20f}$$

Without constraints (20e), this is the well-known p-median problem.[3,24,25] The additional constraints (20e) are capacity constraints that take into account the amount of warehouse space that is actually available to serve product k. Problem (20) is solved for each Type 4 product in turn. If there is more than one Type 4 product, the available capacities $\tilde{B}_i$ are updated each time (20) is solved.

To solve problem (20), we apply a Lagrangian relaxation technique. Nonnegative Lagrange multipliers μ_i are associated with constraints (20e) and unrestricted multipliers λ_j with constraints (20c). These constraints are incorporated into the objective function to produce the relaxed problem:

$$\min \sum_{i\in G} \sum_{j=1}^{J} C^4_{ijk} u_{ijk} + \sum_{j=1}^{J} \lambda_j \left(\sum_{i\in G} u_{ijk} - 1 \right) + \sum_{i\in G} \mu_i \left(\sum_{j=1}^{J} S^4_{jk} u_{ijk} - \tilde{B}_i \right) \quad (21a)$$

subject to

$$\sum_{i\in G} v_{ik} \leq p_k, \quad (21b)$$

$$u_{ijk} \leq v_{ik} \quad \text{for} \quad j = 1, \cdots, J \quad \text{and} \quad i \in G, \quad (21c)$$

$$u_{ijk}, v_{ik} \in \{0, 1\} \quad \text{for} \quad j = 1, \cdots, J \quad \text{and} \quad i \in G. \quad (21d)$$

The relaxed problem can be solved by inspection. For each DC i, compute

$$R_i = \sum_{j=1}^{J} \min(C^4_{ijk} + \lambda_j + \mu_i S^4_{jk}, O). \quad (22)$$

This represents the contribution to the objective function (21a) that is possible if $v_{ik} = 1$. For given λ_j and μ_i, it is optimal in (21) to choose those DCs corresponding to the p_k smallest values of R. (If $v_{ik} = 1$, then it is optimal to choose $u_{ijk} = 1$ only if $C^4_{ijk} + \lambda_j + \mu_i S^4_j \leq O$.)

If these optimal values of u_{ijk} satisfy constraints (20e) and (20c), then we obtain a solution to the original problem (20). If not, then the values of λ_j and μ_i are modified for the next iteration. If $\sum_{i\in G} u_{ijk} > 1$, then λ_j is increased, whereas if $\sum_{i\in G} u_{ijk} < 1$, λ_j is decreased. Likewise, if $\sum_{j=1}^{J} S^4_{jk} u_{ijk} > \tilde{B}_i$, then μ_i is increased. To avoid oscillations, we do not allow values of μ_i to decrease. Thus, if $\sum_{j=1}^{J} S^4_{jk} u_{ijk} \leq \tilde{B}_i$, the constraint is satisfied and μ_i is held fixed. Procedures for updating these multipliers are analogous to those described in Section 3.3.

If a feasible solution is not found within a reasonable number of iterations (again, 40 has been used successfully), then a solution is generated based on the last set G_k of R_i values. The DCs corresponding to the p_k smallest R_i values are chosen to serve product k (i.e., for these DCs set $v_{ik} = 1$). For each demand area j, we find index e such that

$$C^4_{ejk} + \mu_e S^4_{jk} = \min_{i\in G_k} \{C^4_{ijk} + \mu_i S^4_{jk}\}, \quad (23)$$

where the μ_i are also taken from the last iteration. Then, for each j, set

$$u_{ijk} = \begin{cases} 1 \text{ if } i = e \\ 0 \text{ otherwise} \end{cases} \quad \text{for } i \in G. \tag{24}$$

No immediate modification of this solution is made, even in the event that this solution violates capacity constraints (20e). The solution will be adjusted to feasibility later in the facility sizing routine (see Section 3.5). In the meantime, the amount of violation is small, since Type 4 products are assumed to constitute only a small portion of the demand.

The description of the algorithm above can be modified in a straightforward way so that the special DCs are restricted to be chosen from among a predetermined list of DCs for each special product. If none of the facilities from the predetermined list were to appear in the solution obtained in the primary optimization routine, then the model would serve demand areas directly from the Type 4 product vendors. This possibility could be completely avoided by fixing one or more of the facilities on the list to be open a priori.

3.5 LDC locations

After a set of open DCs and demand area assignments are chosen, both for Type 4 products (Section 3.3) and other products (Section 3.2), we attempt to introduce LDCs into the solution. First, we determine a tentative set of open LDC locations and accompanying demand area assignments. With LDCs in place, utilized capacity in some DCs may now be decreased, thus allowing additional demand areas to be assigned. We then check if any "small" DCs can now be closed advantageously and their demand areas reassigned.

3.5.1 Determining tentative LDC assignments

In this routine, we first order the DCs in terms of decreasing throughput (i.e., amount of warehouse space occupied). For each DC in order, we perform an "LDC assignment procedure" to determine the set of LDCs that should be associated with the DC. We consider the larger DCs first, since they are more likely to serve a larger geographic region and, hence, more likely to benefit from the presence of subsidiary LDC locations. The following is the LDC assignment procedure for DC i:

Step 1. Let Q_i denote the set of demand areas j that are assigned to DC i. Let E denote the set of possible LDC locations that have not already been assigned to another DC. For each $j \in Q_i$, determine the LDC $e \in E$ that satisfies

$$A_{ije}^2 = \min_{\ell \in E} A_{ij\ell}^2. \tag{25}$$

For each such $j \in Q_i$, in turn, make a tentative assignment of j to e if

$A^2_{ije} < A^1_{ij}$, unless the assignment would result in the violation of capacity bound $\bar{B}^2_e$. If assigning j to a particular LDC e would violate the capacity bound, check other demand areas that are already assigned to e; if a feasible solution with lower total cost can be obtained by removing another demand area from LDC e and replacing it with demand area j, then do so. Let P_ℓ denote the set of demand areas j tentatively assigned to LDC ℓ at the end of step 1.

Step 2. Consider those LDC locations to which demand areas $j \in Q_i$ have been tentatively assigned at the end of step 1. Sort these LDCs in order of increasing throughput. Thus, underutilized LDCs, which are less likely to be cost-effective, are considered first.

Step 3. Consider each LDC ℓ on the list in order. If the throughput due to tentative assignments is greater than some threshold amount (say, some fraction of the minimum capacity B^2_ℓ), leave its assignments unchanged. Otherwise, remove the LDC from the list, cancel the tentative demand area assignments to LDC ℓ, and attempt to reassign the demand areas $j \in P_\ell$, if possible, to other LDCs. (That is, find another LDC $\bar{e}$ such that

$$A^2_{ij\bar{e}} = \min_{\substack{\ell \in E \\ \ell \neq e}} A^2_{ij\ell}. \tag{26}$$

If $A^2_{ij\bar{e}} < A^1_{ij}$, tentatively assign j to $\bar{e}$.)

Step 4. Resort LDCs remaining on the list in order of increasing throughput.

Step 5. For each LDC ℓ remaining on the list, estimate the total cost for serving demand areas $j \in P_\ell$ using DC i and LDC ℓ. This estimate includes the assignment costs $\sum_{j\in P_\ell} A^2_{ij\ell}$ plus the fixed setup cost α^2_ℓ, plus variable setup costs $\beta^2_\ell \sum_{j\in P_\ell} S^2_j$, plus the concave operating cost $f^2_\ell \left(\sum_{j\in P_\ell} S^2_j\right)$, and minus the closing cost c^2_ℓ. Compare this with an estimate of the cost for serving demand areas $j \in P_\ell$ using DC i alone. This estimate includes the assignment costs $\sum_{j\in P_\ell} A^1_{ij}$, plus variable setup costs $\beta^1_i \sum_{j\in P_\ell} S^1_j$, plus the difference in the concave operating cost function due to the additional throughput. If the cost using the LDC is less, then open LDC ℓ and make the tentative assignments permanent. If not, then attempt to tentatively reassign the demand areas $j \in P_\ell$ to other LDCs still on the list. □

Obviously, more sophisticated procedures can be designed to decide assignments for LDCs. However, since it is expected that demand areas will only be assigned to LDCs that are relatively proximate, the potential number of economically attractive assignments is limited. Thus, more sophisticated procedures have not been found necessary.

After completing this procedure for each DC i that is open, we have a tentative set of open LDCs and LDC assignments.

3.5.2 Closing small DCs

We allow DCs with relatively small throughput to be closed and their demand areas reassigned. We first sort open DCs in a list in order of increasing throughput. In this way, the smaller DCs that are more likely to close will be considered first. For each DC i on the list, we then execute a "DC closing procedure," which follows.

Step 1. If the throughput of DC i exceeds some minimum threshold (say, half the minimum capacity B_i^1), then leave it as is; go on to the next DC. If not, continue with step 2.

Step 2. For each $j \in Q_i$, attempt to tentatively reassign the demand area to another DC. This reassignment should be to a DC that is feasible; that is, the DC should have sufficient spare capacity to handle the demand area, and the demand area should be within the radius of operation for the DC. If there is more than one such feasible DC, choose the one that minimizes the assignment cost for j. If, for some $j \in Q_i$, no feasible reassignment is possible, then cancel all tentative reassignments, keep DC i open, and go on to the next DC. Otherwise, continue with step 3.

Step 3. Compare the additional assignment cost, operating cost and facility closing cost brought on by reassigning demand areas $j \in Q_i$. Compare this with the savings in setup cost for DC i. If it is advantageous to close DC i, make the tentative reassignments permanent. Otherwise, cancel all tentative reassignments and go on to the next DC. □

At this point, the set of open DCs and LDCs is determined. There remains only the question of sizing these open facilities, which we address in the next subsection.

3.6 Facility sizing

The model must determine the number q_i^1 of increments installed such that $B_i^1 + b_i^1 q_i^1 \leq \bar{B}_i^1$. At this point, there is a tentative set of demand area assignments that require amount σ_i^1 [see (8g)] of warehouse space at each DC i. The simplest sizing routine would be to choose q_i^1 to be the smallest integer such that $\sigma_i^1 \leq B_i^1 + b_i^1 q_i^1$. There are two reasons why we might want to modify this approach:

1. DC i is overcapacitated by an amount W_i (perhaps because of assignments that were made for Type 4 products).

2. By moving some demand areas to other DCs, we can perhaps install one less increment of space, thereby saving variable set-up cost $\beta_i^1 b_i^1$. Suppose that if we reduced the requirements for warehouse space at DC i by W_i units, we could install one less increment of space; call this value W_i the "excess" space requirement.

The facility sizing routine attempts to adjust demand area assignments in order to allow certain facilities to be installed at a smaller capacity. It begins by ordering the DCs in a list. First, any overcapacitated DCs are entered in the list. Next, other DCs are entered in the list in order of increasing W_i. For each DC on this list, in order, we then perform the following "DC sizing routine."

Step 1. For each demand area $j \in Q_i$ find its next best "feasible" DC assignment and compute the associated assignment cost differential. (By feasible we mean that assigning the demand area to the DC would not cause certain capacity limits to be exceeded. If DC i, the facility being sized, is overcapacitated, we take this limit to be the maximum capacity of the other DC; otherwise, we take it to be the capacity size required to serve the current assignments to the other DC. Assign an arbitrarily large cost differential if no other feasible assignment is possible.) At this point, consider only reassignments to a DC alone. The use of LDCs will be considered in step 5 below.

Step 2. Sort demand areas $j \in Q_i$ in order of increasing cost differential.

Step 3. Tentatively reassign a sufficient number of demand areas from the top of the list so as to decrease the throughput at DC i by an amount greater than or equal to the excess W_i.

Step 4. If DC i is overcapacitated, make permanent the reassignments found in step 3. If not, check the cost differentials for the reassignments. If the sum of the cost differentials for the reassigned demand areas is greater than the savings $\beta_i^1 b_i^1$ in variable setup cost, then cancel the reassignments. Otherwise, make the reassignments permanent.

Step 5. If demand area reassignments were made permanent in step 4, then consider the possible use of LDCs for each such demand area. In particular, if demand area j were permanently reassigned to a DC i', examine those LDCs ℓ that are now associated with DC i' (i.e., $z_{i'\ell} = 1$). Determine if total costs can be reduced by assigning demand area j to one of these LDCs. If so, assign j to the LDC that results in the minimum total cost. □

After this procedure is completed, the values of q_i^1 (number of space increments for DCs) and q_ℓ^2 (number of space increments for LDCs) are chosen to be the smallest integers that provide sufficient warehousing space to handle the assigned demand areas.

IV. IMPLEMENTATION DETAILS AND CONCLUDING REMARKS

To provide an effective tool for decision makers, our model was designed to be flexible and efficient. Flexibility in the implementation allows the user to analyze many different real situations using the same "generic" model. Further flexibility comes from an implementation that permits some variables to be fixed a priori, thus allowing the user to impose various "nonquantifiable" conditions on the model. For example, various DCs or LDCs can be fixed open or closed. Certain DCs can be forbidden from handling Type 4 products; certain demand areas can be assigned a priori to a given DC or LDC. These conditions can be imposed to reflect some physical constraint or corporate policy. Such conditions can also be imposed after studying a previous solution obtained from the model. In this way, the model is "forced" to consider alternate solutions of interest to the user. The user may also wish to perform a sensitivity analysis in which the model is run several times with variations in one or more cost parameters.

Efficiency of the algorithm is essential so that multiple runs as described above can be accomplished in a short time frame without excessive computation. The current implementation allows for a maximum of 10 different products, 50 possible DC locations, 40 possible LDC locations, and 200 demand areas. Some average run times for a variety of problems that have been encountered in practice are given in Table I. (Problem I, in particular, was the basis for a nationwide distribution planning study.) These times were obtained on an Amdahl 470/V8 operating under MVS; the code was compiled using the FORTRAN 77 compiler. All times are within an acceptable range for performing multiple runs in an economic study. Note that, as is typical in combinatorial optimization problems, run times can vary among different problems of the same size. For example, problem H is smaller than problem E, but took over twice the CPU time (262.2 seconds versus 115.6 seconds). (This variation is due primarily to the difference in the number of iterations required by the Kelly and Khumawala algorithm, which is used within the primary optimization procedure described in Section 3.3.)

The many cost components considered (including nonlinear operating costs), the ability to incorporate such features as different

Table I—Average execution times

Number of Problem	Number of Demand Areas	Number of DCs	Number of LDCs	Number of Special Products	CPU Seconds
A	30	8	10	1	4.6
B	54	8	25	0	11.1
C	149	8	25	1	16.7
D	54	48	25	0	61.2
E	100	50	20	1	115.6
F	100	48	25	0	126.6
G	149	48	25	0	170.1
H	100	50	0	0	262.2
I	158	50	0	0	603.7

product types and capacity bounds and subsidiary warehouses, the flexibility offered by the ability to fix variables a priori, and the efficiency in run times make this model a unique and useful tool. It should provide genuine insight when used to study existing or proposed material logistics systems.

V. ACKNOWLEDGMENTS

Hanan Luss provided very valuable advice and support throughout the development of this model. His suggestions greatly improved the solution algorithm and its presentation here. Patty Donohue developed the user interface routines and provided essential help and support during the implementation and testing of the program. Dave Aff, Marv Berkowitz, and Steve Waters contributed to many helpful discussions concerning the model formulation and data requirements.

REFERENCES

1. R. L. Francis, L. F. McGinnis, and J. A. White, "Locational Analysis," Eur. J. Oper. Res., *12* (March 1983), pp. 220–52.
2. G. Y. Handler and P. B. Mirchandani, *Location on Networks*, Cambridge: M.I.T. Press, 1979.
3. B. C. Tansel, R. L. Francis, and T. J. Lowe, "Location on Networks: A Survey, Part I: The p-Center and p-Median Problems," Manage. Sci., *29* (April 1983), pp. 482–97.
4. M. A. Efroymson and T. L. Ray, "A Branch-Bound Algorithm for Plant Location," Oper. Res., *14* (May–June 1966), pp. 361–8.
5. D. Erlenkotter, "A Dual-Based Procedure for Uncapacitated Facility Location," Oper. Res., *14* (November–December 1978), pp. 992–1009.
6. B. M. Khumawala, "An Efficient Branch-and-Bound Algorithm for the Warehouse Location Problem," Manage. Sci., *18* (August 1972), pp. 718–31.
7. A. A. Kuehn and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," Manage. Sci., *9* (July 1963), pp. 643–66.
8. G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey, "The Uncapacitated Facility Location Problem," Manage. Sci. Res. Rep. No. MSRR 493, Carnegie-Mellon University, August 1983.
9. J. Krarup and P. Pruzan, "Simple Plant Location Problem: Survey and Synthesis," Eur. J. Oper. Res., *12* (January 1983), pp. 36–81.
10. G. T. Ross and R. M. Soland, "Modeling Facility Location Problems as Generalized Assignment Problems," Manage. Sci., *24* (November 1977), pp. 345–57.
11. M. L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problems," Manage. Sci., *27* (January 1981), pp. 1–8.
12. P. S. Davis and T. L. Ray, "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem," Nav. Res. Logis. Quart., *16*, (September 1969), pp. 331–4.
13. A. M. Geoffrion and R. D. McBride, "Lagrangean Relaxation Applied to Capacitated Facility Location Problems," AIIE Trans., *10* (March 1978), pp. 40–7.
14. R. M. Nauss, "An Improved Algorithm for the Capacitated Facility Location Problem," J. Oper. Res. Soc., *29* (December 1978), pp. 1195–201.
15. D. L. Kelly and B. M. Khumawala, "Capacitated Warehouse Location With Concave Costs," J. Oper. Res. Soc., *33* (September 1982), pp. 817–26.
16. B. M. Khumawala and D. L. Kelly, "Warehouse Location With Concave Costs," INFOR, *12* (February 1974), pp. 55–65.
17. R. M. Soland, "Optimal Plant Location With Concave Costs," Oper. Res., *22* (March–April 1974), pp. 373–85.
18. A. M. Geoffrion and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition," Manage. Sci., *20* (January 1974), pp. 822–44.
19. D. Erlenkotter, "A Comparative Study of Approaches to Dynamic Location Problems," Eur. J. Oper. Res., *6* (February 1981), pp. 133–43.

20. H. Luss, "Operations Research and Capacity Expansion Problems: A Survey," Oper. Res., *30* (September–October 1982), pp. 907–47.
21. C. L. Monma and M. Segal, "A Primal Algorithm for Finding Minimum-Cost Flows in Capacitated Networks With Applications," B.S.T.J., *61* (July–August 1982), pp. 949–68.
22. M. Held, P. Wolfe, and H. Crowder, "Validation of Subgradient Optimization," Math. Program., *6* (January 1974), pp. 62–88.
23. H. M. Salkin, *Integer Programming*, Reading, MA: Addison-Wesley, 1975.
24. S. C. Narula, U. I. Ogbu, and H. M. Samuelsson, "An Algorithm for the p-Median Problem," Oper. Res., *25* (July–August 1977), pp. 709–12.
25. S. L. Hakimi, "Optimal Locations of Switching Centers and the Absolute Centers and Medians of a Graph," Oper. Res., *12* (May–June 1964), pp. 450–9.

AUTHOR

John G. Klincewicz, S.B. (Mathematics), 1975, The Massachusetts Institute of Technology; M.A., 1977, and Ph.D. (Operations Research), 1979, Yale University; AT&T Bell Laboratories, 1979—. At AT&T Bell Laboratories, Mr. Klincewicz is a member of the Operations Research Department. His research interests include applications of mathematical programming and the development of algorithms for network flow problems and facility location problems. Member, Operations Research Society, Mathematical Programming Society.